# 路云展示系统API文档

目录

[路云展示系统API文档 1](#_Toc195652002)

[1. 用户管理 3](#_Toc195652003)

[1.1. 用户登录 3](#_Toc195652004)

[1.2. 用户注册 4](#_Toc195652005)

[1.3. 查询用户列表(分页和条件过滤) 5](#_Toc195652006)

[1.4 查询用户详细 6](#_Toc195652007)

[1.5 修改用户 7](#_Toc195652008)

[1.6 删除用户 8](#_Toc195652009)

[1.7 用户密码重置 9](#_Toc195652010)

[1.8 用户密码修改 9](#_Toc195652011)

[1.9 新建用户 10](#_Toc195652012)

[1.10 用户数据导出 11](#_Toc195652013)

[2. Mqtt订阅管理 12](#_Toc195652014)

[2.1 获取Mqtt配置、主题、状态信息 13](#_Toc195652015)

[2.2 设置Mqtt配置、主题信息 14](#_Toc195652016)

[2.3 开始订阅 15](#_Toc195652017)

[2.5 停止订阅 16](#_Toc195652018)

[2.6 分页查询信息 17](#_Toc195652019)

[2.7 测试连接是否成功 18](#_Toc195652020)

[2.8 根据主题、时间范围导出消息列表 19](#_Toc195652021)

[3. 数据管理 20](#_Toc195652022)

[3.1 车辆信息 20](#_Toc195652023)

[3.1.1 查询所有车辆数据 (测试接口，前端可不阅) 20](#_Toc195652024)

[3.1.2 根据vehicleId模糊查询 22](#_Toc195652025)

[3.1.2.1 车辆信息总览(Dashboard) 24](#_Toc195652026)

[3.1.3 导出车辆信息 26](#_Toc195652027)

[3.1.4 删除车辆信息根据stateId 27](#_Toc195652028)

[3.1.5 删除车辆信息根据vehicleId 28](#_Toc195652029)

[3.1.6 查询某车辆某时间段的数据 29](#_Toc195652030)

[3.2 Rcu 30](#_Toc195652031)

[3.2.1 分页查询rcu设备信息 30](#_Toc195652032)

[3.2.2 根据rcuid和时间范围分页查询对象列表 31](#_Toc195652033)

[3.2.3 根据objsflag分页查询对象详细数据 33](#_Toc195652034)

[3.2.4 根据rcuid删除该设备以及其相关所有信息 34](#_Toc195652035)

[3.2.5 根据objsflag删除该设备以及其相关所有信息 35](#_Toc195652036)

[3.2.6 根据objsflag和objId删除某条rcu信息中的某个对象信息 36](#_Toc195652037)

[3.2.7 根据rcuids和时间范围导出objs数据 37](#_Toc195652038)

[3.2.8 根据objsflags和时间范围导出objs数据 38](#_Toc195652039)

[4. 操作日志管理 38](#_Toc195652040)

[4.1 分页查询用户操作日志 38](#_Toc195652041)

[4.2 删除数据库日志数据 41](#_Toc195652042)

[4.3 导出数据库日志数据 42](#_Toc195652043)

[5. 系统参数管理 42](#_Toc195652044)

[5.1 分页查询系统参数列表 42](#_Toc195652045)

[5.2 删除参数数据 44](#_Toc195652046)

[5.3 修改参数 45](#_Toc195652047)

[5.4 新增参数 45](#_Toc195652048)

[6. 首页界面 46](#_Toc195652049)

[6.1 交通设备数量总览-查询 46](#_Toc195652050)

[6.2 设备健康状态分布-查询 47](#_Toc195652051)

[6.3 查询异常设备（告警、离线） 48](#_Toc195652052)

[6.4 根据RcuId查询设备信息 49](#_Toc195652053)

[6.5 根据车辆id查询车辆信息 51](#_Toc195652054)

[6.6 拓扑图填充 52](#_Toc195652055)

[6.7 首页流量分析曲线图 55](#_Toc195652056)

[6.8 首页根据时间范围统计车辆类型分布-饼图 56](#_Toc195652057)

[6.9 获取交通事件 58](#_Toc195652058)

[6.10 统计人、车数量和差值 59](#_Toc195652059)

[6.11 地标路段实时流量排名 60](#_Toc195652060)

[6.12 分段统计车辆类型以及数量列表 61](#_Toc195652061)

[6.13 统计车辆速度分布区间和数量 62](#_Toc195652062)

[6.14 分页查询设备列表 63](#_Toc195652063)

[6.15 分页查询车辆列表 65](#_Toc195652064)

[7. 详细页界面 66](#_Toc195652065)

[7.1 详细页流量分析曲线图 66](#_Toc195652066)

[7.2 详细页流量分析条形图 67](#_Toc195652067)

[7.3 查询所有可建模数据--Blender相关 68](#_Toc195652068)

[7.4 导出建模数据 71](#_Toc195652069)

[7.5 建模数据转模型文件 72](#_Toc195652070)

[7.6 根据RcuId路口信息 72](#_Toc195652071)

[8. RBAC权限管理 73](#_Toc195652072)

[8.1 查询所有角色 73](#_Toc195652073)

[8.2 创建角色 76](#_Toc195652074)

[8.3 删除角色 77](#_Toc195652075)

[8.4 查询所有权限 78](#_Toc195652076)

[8.5 增加角色权限 79](#_Toc195652077)

[8.6 删除角色权限 80](#_Toc195652078)

# 用户管理

## 用户登录

**接口描述**

登录RoadCloudVisualizationSystem系统的时候需要输入用户名和密码进行登录，不登录不能访问系统。

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /login

Body中放置请求参数，参数详情如下：

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| username | 是 | Varchar |  | 用户登录时所需用户名 |
| password | 是 | Varchar |  | 用户登录时所需密码 |

**返回说明**

返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 是否必选 |  | 类型 | 说明 |
| status | 是 |  | Varchar | 用于提示用户登录成功或者失败  success或者fail |
| msg | 否 |  | Varchar | 用于提示用户登录失败的原因   1. 用户名不存在 2. 密码错误 3. 网络连接失败 |
| Authorization | 否 |  | Varchar | 登录成功后返回token，失败不返回  前端需要保存该token，携带时加上前缀“Bearer ”（有一个空格） |

**返回示例**

登录成功

{

"status": “success”,

"Authorization": "eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiI1MzQ2MTk1Ny0zZDcwLTQxZTAtYThmNi1lZTM4NWMzMTkyNzIiLCJzdWIiOiJsaXNpIiwiaXNzIjoic2ciLCJpYXQiOjE3NDQwNzc5NTEsImV4cCI6MTc0NDA4NTE1MX0.D\_KdYW8Uh1BwNs5wD5vFk4xiePpz-zLts\_9\_6FJp1WI"

}

登录失败

{

"status": “fail”,

“msg”: “密码错误”

}

## 用户注册

**接口描述**

本接口实现了用户注册的功能，用户通过向服务器写入自己的用户数据信息，从而实现登录以及后续系统其他功能的使用。

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /register

Body中放置请求参数，参数详情如下：

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| uid | 是 | Varchar |  | 用户不可自定义，后端自行分配 |
| username | 是 | Varchar |  | 用户自定义的新用户名 |
| password | 是 | Varchar |  | 用户自定义的密码 |
| authority | 是 | Varchar |  | 默认值为1(普通用户)后端自行分配 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示用户注册成功或者失败  1. success  2. fail |
| msg | 否 | Varchar | 用于提示用户注册失败的原因   1. 用户名重复 2. 网络连接失败 |

**返回示例**

注册成功

{

"status": “success”

}

注册失败

{

"status": “fail”,

“msg”: “网络连接失败”

}

## 查询用户列表(分页和条件过滤)

**接口描述**

本接口实现了用户查询的功能，管理员在后台管理系统切换到用户管理后会分页显示当前存储的用户，以便对用户进行增删改查；并支持通过用户名作为过滤条件。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/user/list

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| username | 否 | Varchar |  | 用户名，作为筛选条件 |
| pageNum | 是 | int |  | 当前页数 |
| pageSize | 是 | Int |  | 每页记录数 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示查询成功或者失败  1. success  2. fail |
| total | 否 | Int | 用户总条数 |
| rows | 否 | List<User> | 返回查询到的所用户列表 |

**返回示例**

查询成功

{

"status": “success”,

“total”: 1

“rows”: [

{“uid”: “lisi\_1743995092579”,

“username”: “lisi”,

“password”: “ssadfasfsa”,

“authority”: “”

}

]

}

查询失败

{

"status": “fail”,

}

## 查询用户详细

**接口描述**

本接口实现了查询具体用户的功能。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/user/

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| uid | 是 | Varchar |  | 用户名筛选条件 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示查询成功或者失败  1. success  2. fail |
| data | 否 | User | 返回查询到的用户 |

**返回示例**

查询成功

{

"status": “success”,

“data”: {“uid”: “lisi\_1743995092579”,

“username”: “lisi”,

“password”: “ssadfasfsa”,

“authority”: “”

}}

查询失败

{

"status": “fail”,

}

## 修改用户

**接口描述**

本接口实现了修改具体用户的功能。

**请求说明**

请求示例

HTTP 方法: PUT

请求URL: /system/user/

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| uid | 是 | String |  | 用户id |
| username | 是 | String |  | 用户名 |
| password | 否 | String |  | 用户密码 |
| authority | 是 | String |  | 用户角色 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示查询成功或者失败  1. success  2. fail |
| msg | 是 | Varchar | 1. 修改成功 2. 修改失败 3. 修改失败，用户名重复 |

**返回示例**

修改成功

{

"status": “success”,

“msg”：“修改成功”

}

修改失败

{

"status": “fail”,

“msg”：“修改失败”

}

## 删除用户

**接口描述**

本接口实现了删除具体用户的功能。

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/user/

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| uid | 是 | String |  | 用户id |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示查询成功或者失败  1. success  2. fail |
| msg | 是 | Varchar | 1. 删除成功 2. 删除失败 |

**返回示例**

删除成功

{

"status": “success”,

“msg”：“删除成功”

}

修改失败

{

"status": “fail”,

“msg”：“删除失败”

}

## 用户密码重置

**接口描述**

本接口实现了用户密码重置的功能，情景是当用户忘记了密码，需要传给后端用户id和新密码。

**请求说明**

请求示例

HTTP 方法: PUT

请求URL: /system/user/resetPwd

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| uid | 是 | String |  | 用户id |
| password | 是 | String |  | 新密码 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示查询成功或者失败  1. success  2. fail |
| msg | 是 | Varchar | 1. 重置密码成功 2. 重置密码失败 |

**返回示例**

重置密码成功

{

"status": “success”,

“msg”：“重置密码成功”

}

重置密码失败

{

"status": “fail”,

“msg”：“重置密码失败”

}

## 用户密码修改

**接口描述**

本接口实现了用户密码修改的功能，需要用户输入旧密码和新密码来进行修改。

**请求说明**

请求示例

HTTP 方法: PUT

请求URL: /system/user/updatePwd

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| uid | 是 | String |  | 用户id |
| oldPassword | 是 | String |  | 旧密码 |
| newPassword | 是 | String |  | 新密码 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示查询成功或者失败  1. success  2. fail |
| msg | 是 | Varchar | 1. 修改密码成功 2. 修改密码失败，旧密码错误 3. 修改密码失败，密码不能为空 |

**返回示例**

修改密码成功

{

"status": “success”,

“msg”：“修改密码成功”

}

修改密码失败

{

"status": “fail”,

“msg”：“修改密码失败，旧密码错误或者为空”

}

## 新建用户

**接口描述**

本接口实现了新建用户的的功能，需要输入用户名、密码和权限等级。

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /system/user

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| username | 是 | String |  | 用户名 |
| password | 是 | String |  | 密码 |
| authority | 是 | String |  | 权限等级 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示成功或者失败  1. success  2. fail |
| msg | 是 | Varchar | 1. 新建用户成功 2. 新建用户失败，用户名已存在 3. 新建用户失败，服务器内部问题 4. 新建用户失败，信息填写不完整 |

**返回示例**

新建用户成功

{

"status": “success”,

“msg”：“新建用户成功”

}

新建用户失败

{

"status": “fail”,

“msg”：“新建用户失败，用户名已存在”

}

## 1.10 用户数据导出

**接口描述**

本接口实现导出所有用户的功能，所有导出的用户数据将存放在应该excel表中，返回一个excel表。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/user/export

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示导出失败  1. fail |
| msg | 是 | Varchar | 1. 用户数据导出失败，网络或者服务器错误 |

**返回示例**

导出成功

HTTP/1.1 200 OK

Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

Content-Disposition: attachment; filename=users.xlsx

[二进制Excel文件数据]

导出失败

HTTP/1.1 500 Internal Server Error

Content-Type: application/json

{

"status": "fail",

"msg": "导出失败: 网络或者服务器错误",

}

# Mqtt订阅管理

状态有“正在进行订阅”、“未进行订阅”，可以的话设置禁用的按钮（比如在订阅时，不能点击“开始订阅”；“未进行订阅”时，不能点击“停止订阅按钮”）；请限制只能在“未进行订阅”时才能修改配置信息并提交，或者在“正在进行订阅”状态时，通过点击“停止订阅”先停止，再修改配置信息。

点击开始订阅、修改配置、停止订阅的按钮时，应该要有应弹窗提醒（等待后端的响应），因为如果后端网络状态无法连接时会等待较长时间。

## 获取Mqtt配置、主题、状态信息

**接口描述**

本接口实现了获取Mqtt当前配置、主题、状态的功能，状态有“正在进行订阅”、“未进行订阅”，可以的话设置禁用的按钮（比如在订阅时，不能点击“开始订阅”；“未进行订阅”时，不能点击“停止订阅按钮”）；请限制只能在“未进行订阅”时才能修改配置信息并提交，或者在“正在进行订阅”状态时，通过点击“停止订阅”先停止，再修改配置信息。

点击开始订阅、修改配置、停止订阅的按钮时，应该要有应弹窗提醒（等待后端的响应），因为如果后端网络状态无法连接时会等待较长时间。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/mqtt

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示查询成功或者失败  1. success  2. fail |
| msg | 是 | Varchar | 1. 获取成功 2. 获取失败，服务器内部错误 |
| currentConfig | 是 | MqttConfig | 配置信息：包括brokerUrl、clientId、topics、username  、password |
| isRunning | 是 | boolean | 是否正在运行（订阅） |

**返回示例**

获取成功

{

"status": “success”,

“msg”：“获取成功”,

“currentConfig”: {

“brokerUrl”: "tcp://192.168.31.250:1887",

“clientId”: “myclient”,

“topics”: [

{“topic”: “frsu/#”,“qos”: 2},

{"topic": "fvpub/obu/state/#", "qos": 2},

{"topic": "frpub/rcu/#", "qos": 2}

]

“username”: “smqtt”

“password”: “smqtt”

}

}

获取失败

{

"status": “fail”,

“msg”：“获取失败，服务器内部错误”

}

## 设置Mqtt配置、主题信息

**接口描述**

本接口实现了对Mqtt配置、主题信息的设置，需要传入brokerUrl、clientId、topics、username、password信息。

只能在“未进行订阅”状态时才能设置，或者停止订阅。

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /system/mqtt

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| brokerUrl | 是 | String |  | 请求地址 |
| clientId | 是 | String |  | 客户端id |
| topics | 是 | List<TopicSubscription> |  | 主题信息 |
| username | 是 | String |  | 用户名 |
| password | 是 | String |  | 密码 |

TopicSubscription数据结构说明：

String topic

int qos（限制0，1，2）

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示设置是否成功  1. success  2. fail |
| msg | 是 | Varchar | 1. 配置设置成功 2. 配置设置失败，请先停止订阅 3. 配置设置失败，服务器内部错误 |

**返回示例**

获取成功

{

"status": “success”,

“msg”：配置设置成功

}

获取失败

{

"status": “fail”,

“msg”：“获取失败，服务器内部错误”

}

## 开始订阅

**接口描述**

本接口实现对Mqtt的开始订阅功能，只能在“订阅未进行”状态情况下才能开始订阅。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/mqtt/start

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示是否成功  1. success  2. fail |
| msg | 是 | Varchar | 1. 开启订阅成功 2. 开启订阅失败，网络连接不到mqtt服务器 3. 开启订阅失败，请先停止订阅 4. 开启订阅失败，配置信息不存在 5. 开启订阅失败，用户密码信息认证无效 |

**返回示例**

开启成功

{

"status": “success”,

“msg”：“开启订阅成功”

}

开启失败

{

"status": “fail”,

“msg”：“开启订阅失败，网络连接不到mqtt服务器”

}

## 停止订阅

**接口描述**

本接口实现对Mqtt的停止订阅功能

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/mqtt/stop

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示是否成功  1. success  2. fail |
| msg | 是 | Varchar | 1. 停止订阅成功 2. 当前未开启订阅 3. 停止订阅失败， 服务器内部错误 |

**返回示例**

停止成功

{

"status": “success”,

“msg”：“当前未开启订阅”

}

停止失败

{

"status": “fail”,

“msg”：“停止订阅失败， 服务器内部错误”

}

## 分页查询信息

**接口描述**

后台管理系统中可以分页查询到历史订阅到的消息以及主题，支持通过topic、datetime来进行过滤。其中topic可选，时间范围必选，请前端默认时间范围为当前5分钟内。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/mqtt/list

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| topic | 否 | String |  | 主题 |
| startTime | 是 | String |  | 开始时间 |
| endTime | 是 | String |  | 结束时间 |
| pageNum | 是 | Integer |  | 页码 |
| pageSize | 是 | Integer |  | 每页数量 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于是否成功  1. success  2. fail |
| total | 否 | Int | 用户总条数 |
| rows | 否 | List<Message> | 返回查询到的所消息列表 |

**返回示例**

查询成功

{

"status": “success”,

“total”: 15

“rows”: [

{“messageId”: “6”,

“topic”: “frce/#”,

“content”: “{header:…}”,

“receivedTime”: “1743592399252”

},

{“message\_id”: “7”,

“topic”: “frce/rrr/#”,

“content”: “{header:…}”,

“receivedTime”: “1743592399252”

}

]

}

查询失败

{

"status": “fail”,

}

## 测试连接是否成功

**接口描述**

后台管理系统中可以通过当前的配置信息来测试是否能够成功连接到指定的mqtt服务器。

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /system/mqtt/testConnection

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| brokerUrl | 是 | String |  | 请求地址 |
| clientId | 是 | String |  | 客户端id |
| topics | 是 | List<TopicSubscription> |  | 主题信息 |
| username | 是 | String |  | 用户名 |
| password | 是 | String |  | 密码 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于是否成功  1. success  2. fail |
| msg | 否 | Int | 1. 配置信息不完整 2. 连接测试成功 3. 连接测试失败 |

**返回示例**

连接测试成功

{ "status": "success", "msg": "连接测试成功" }

连接测试失败

{ "status": "fail", "msg": "连接测试失败" }

## 根据主题、时间范围导出消息列表

**接口描述**

后台管理系统中可以根据topic（可选），时间范围（必选）导出消息列表。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/mqtt/export

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| topic | 否 | String |  | 主题 |
| startTime | 是 | String |  | 开始时间 |
| endTime | 是 | String |  | 结束时间 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于是否成功  1. success  2. fail |
| filepath | 否 | Varchar | 路径 |

**返回示例**

导出成功

{

    "filepath": "/exports/message\_wangfeng1744902966934.xlsx",

    "status": "success"

}

导出失败

{

"status": "fail",}

# 数据管理

## 车辆信息

### 查询所有车辆数据 (测试接口，前端可不阅)

**接口描述**

本接口实现了直接查询数据库数据并分页返回，可以通过分页的方式进行查询

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/vehicle/list

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| pagaNum | 否 | String | 从1开始 | 为空默认为1 |
| pageSize | 否 | String |  | 为空默认为10 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | success 成功  fail 失败 |
| total | 否 | String | 数据库总数据量 |
| list | 否 | list | 存放车辆的字段组 |

**返回示例**

成功查询

{

"total": 77,

"rows": [

{

"elevation": "2802",

"heading": "358.8875",

"stateId": "QD1E000J\_1723504680233",

"velocityGNSS": "0.0",

"latitude": "29.6264923",

"timestampGNSS": "1723504680233",

"vehicleId": "QD1E000J",

"longitude": "106.3337938"

},

{

"elevation": "2802",

"heading": "358.8875",

"stateId": "QD1E000J\_1723504680333",

"velocityGNSS": "0.0",

"latitude": "29.6264921",

"timestampGNSS": "1723504680333",

"vehicleId": "QD1E000J",

"longitude": "106.3337938"

}

]

}

查询失败

{

"status": “fail”

}

### 根据vehicleId查询该车辆详细信息

**接口描述**

根据vehicleId进行模糊查询，如果想要精确的具体某个vehicleId的信息，可将vehicleId输入完整以实现精确查询

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/vehicle/list/vehicleId

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| vehicleId | 是 | string |  | vehicleId |
| pagaNum | 是 | String | 从1开始 | 单页数量 |
| pageSize | 是 | String |  | 页码 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | success 成功  fail 失败 |
| total | 否 | String | 数据库总数据量 |
| rows | 否 | list | 存放车辆的字段组 |

**返回示例**

查询成功

{

"total": 77,

"rows": [

{

"elevation": "2802",

"heading": "358.8875",

"stateId": "QD1E000J\_1723504680233",

"velocityGNSS": "0.0",

"latitude": "29.6264923",

"timestampGNSS": "1723504680233",

"vehicleId": "QD1E000J",

"longitude": "106.3337938"

},

{

"elevation": "2802",

"heading": "358.8875",

"stateId": "QD1E000J\_1723504680333",

"velocityGNSS": "0.0",

"latitude": "29.6264921",

"timestampGNSS": "1723504680333",

"vehicleId": "QD1E000J",

"longitude": "106.3337938"

}

]

}

查询失败

{

"status": "fail"

}

### 车辆信息总览(Dashboard)

**接口描述**

将车辆根据vehicleId进行分类，分别包含其对应的位置点数量(position)，以及最后更新时间

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/vehicle/manage

Parma请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| pagaNum | 否 | String | 从1开始 | 为空默认为1 |
| pageSize | 否 | String |  | 为空默认为10 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | success 成功  fail 失败 |
| total | 是 | Int | 数据总条数 |
| rows | 是 | List | 所有车的列表信息 |

**返回示例**

查询成功

{

"total": 5,

"rows": [

{

"count": 15,

"vehicleId": "QD1E000R",

"latestTime": "1723676700778"

},

{

"count": 16,

"vehicleId": "QD1E000J",

"latestTime": "1723504681734"

},

{

"count": 16,

"vehicleId": "QD1E000N",

"latestTime": "1723504530896"

},

{

"count": 15,

"vehicleId": "QD1E000T",

"latestTime": "1723500048348"

},

{

"count": 15,

"vehicleId": "QD1E0026",

"latestTime": "1723485145529"

}

],

"status": "success"

}

查询失败

{

"status": "fail"

}

### 导出车辆信息

**接口描述**

选择起始时间，终止时间，车牌号然后导出车辆信息为xlsx

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/vehicle/export

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| starttime | 是 | string |  | 开始时间(时间戳) |
| endtime | 是 | string |  | 结束时间(时间戳) |
| savedata | 是 | string |  | 要保存的车辆plateNo，多个请用 , 隔开例如渝CG6524,渝E7Y973 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | success 成功  fail 失败 |
| filepath | 否 | string | 保存成功的下载路径 |

**返回示例**

保存成功

{

"status": "fail",

"filepath": "upload/xlsxs/001.xlsx"

}

保存失败

{

"status": "fail"

}

### 删除车辆信息根据stateId

**接口描述**

根据stateId删除单个车辆单个信息

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/vehicle/delete/stateId

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| stateId | 是 | string |  | 车辆单个信息的唯一主键 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | success 成功  fail 失败 |

**返回示例**

删除成功

{

"status": "success"

}

删除失败

{

"status": "fail"

}

### 删除车辆信息根据vehicleId

**接口描述**

根据vehicleId删除车辆信息(多条数据请慎重)

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/vehicle/delete/vehicleId

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| vehicleId | 是 | string |  | 车牌号 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | success 成功  fail 失败 |

**返回示例**

删除成功

{

"status": "success"

}

删除失败

{

"status": "fail"

}

### 查询某车辆某时间段的数据

**接口描述**

根据vehicleId进精确搜索，并且根据starttime和endtime进行过滤

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/vehicle/list/vehicleIdandtime

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| vehicleId | 是 | string |  | vehicleId |
| starttime | 否 | string |  | 默认为负无穷 |
| endttime | 否 | string |  | 默认为正无穷 |
| pagaNum | 是 | String | 从1开始 | 单页数量 |
| pageSize | 是 | String |  | 页码 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | success 成功  fail 失败 |
| total | 否 | String | 数据库总数据量 |
| rows | 否 | list | 存放车辆的字段组 |

**返回示例**

查询成功

{

"total": 77,

"rows": [

{

"elevation": "2802",

"heading": "358.8875",

"stateId": "QD1E000J\_1723504680233",

"velocityGNSS": "0.0",

"latitude": "29.6264923",

"timestampGNSS": "1723504680233",

"vehicleId": "QD1E000J",

"longitude": "106.3337938"

},

{

"elevation": "2802",

"heading": "358.8875",

"stateId": "QD1E000J\_1723504680333",

"velocityGNSS": "0.0",

"latitude": "29.6264921",

"timestampGNSS": "1723504680333",

"vehicleId": "QD1E000J",

"longitude": "106.3337938"

}

]

}

查询失败

{

"status": "fail"

}

## 3.2 Rcu

### 3.2.1 分页查询rcu设备信息

**接口描述**

后台管理系统中可以分页查询到rcu设备的信息，包括rcuId，最新发送数据的时间，通过rcuId可以查询到这个rcu发送过来的对象信息。过滤使用的时间范围请默认为一天内。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/rcu/list

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| rcuId | 否 | String |  | 用于模糊查询 |
| pageNum | 是 | Integer |  | 页码 |
| pageSize | 是 | Integer |  | 每页数量 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于是否成功  1. success  2. fail |
| total | 否 | Int | rcu总条数 |
| rows | 否 | List<RcuObjs> | 返回查询到的所rcu列表 |

**返回示例**

查询成功

{

"status": “success”,

“total”: 15

“rows”: [

{“objsflag”: “FCP0001\_1744021849840”,

“deviceType”: 2,

“rcuId”: “F-CP2027”,

“timestampOfDetOut”: “1743592399252”

},

{“objsflag”: “FCP0001\_1744162995011”,

“deviceType”: 2,

“rcuId”: “F-CP2027”,

“timestampOfDetOut”: “1743592399252”

}

]

}

查询失败

{

"status": “fail”,

}

### 3.2.2 根据rcuid和时间范围分页查询对象列表

**接口描述**

后台管理系统中可需要利用rcuid和时间范围来查找rcu上报的对象数量信息，并后续提供对象详细信息的查询方式。要求前端默认是1h时间内或者自定义。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/rcu/info\_list

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| rcuId | 是 | String |  | Rcu设备id |
| startTime | 是 | String |  | 开始时间 |
| endTime | 是 | String |  | 结束时间 |
| pageNum | 是 | Integer |  | 页码 |
| pageSize | 是 | Integer |  | 每页数量 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于是否成功  1. success  2. fail |
| total | 否 | Int | 数据总条数 |
| rows | 否 | List<Map> | 返回查询到的数据列表 |

**返回示例**

查询成功

{

"status": “success”,

“total”: 15

“rows”: [

{“objsflag”: “F-CP0001\_1744162995011”,

“rcuId”: “F-CP0001”,

"deviceType":2,

"objectiveNum":2,

“timestampOfDetOut”: “1743592399252”,

},

{“objsflag”: “F-CP0001\_1744162995311”,

“rcuId”: “F-CP0001”,

"deviceType":2,

"objectiveNum":2

“timestampOfDetOut”: “1743592399252”,

}

]

}

查询失败

{

"status": “fail”,

}

### 3.2.3 根据objsflag分页查询对象详细数据

**接口描述**

后台管理系统中可需要利用objsflag来分页查询本次消息中的对象详细数据。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/rcu/obj\_list

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| objsflag | 是 | String |  | 唯一标志 |
| pageNum | 是 | Integer |  | 页码 |
| pageSize | 是 | Integer |  | 每页数量 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于是否成功  1. success  2. fail |
| total | 否 | Int | 数据总条数 |
| rows | 否 | List<Obj> | 返回查询到的数据列表 |

**返回示例**

查询成功

{

"status": “success”,

“total”: 2

“rows”: [

{“uuid”: “18f8f1fb5f8f4f5895d87e151ac252a9”,

“objsflag”: “F-CP0001\_1744162995011”,

“longitude”: “106.32563986695847”,

“latitude”: “29.517776039542742”，

"elevation":0,

“heading”: “84.8567368549392”,

"objId":0,

"plateNo":"",

“speed”: “0”,

“type”: “7”

},

{  
 "heading":186.5766, "objId":1,"elevation":0,  
 "latitude":29.5205528, "longitude":106.3162864,  
 "plateNo":"", "speed":0.06, "type":2, "uuid":"4F303030523137343801482821073E2B"  
}

]

}

查询失败

{

"status": “fail”,

}

### 3.2.4 根据rcuid删除该设备以及其相关所有信息

**接口描述**

后台管理系统中可需要利用rcuid删除rcu设备以及其所有信息

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/rcu

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| rcuId | 是 | String |  | rcu唯一标志 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示是否删除成功  1. success  2. fail |
| msg | 否 | Varchar | 1. 删除成功 2. 删除失败 |

**返回示例**

删除成功

{

"status": “success”,

“msg”: “删除成功”

}

删除失败

{

"status": “fail”,

“msg”：“删除失败”

}

### 3.2.5 根据objsflag删除该消息以及其所有对象信息

**接口描述**

后台管理系统中可删除某条rcu发来的所有对象信息

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/rcu/objs

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| objsflag | 是 | String |  | rcu发来的信息唯一标志 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示是否删除成功  1. success  2. fail |
| msg | 否 | Varchar | 1. 删除成功 2. 删除失败 |

**返回示例**

删除成功

{

"status": “success”,

“msg”: “删除成功”

}

删除失败

{

"status": “fail”,

“msg”：“删除失败”

}

### 3.2.6 根据objsflag和objId删除某条rcu信息中的某个对象信息

**接口描述**

后台管理系统中可删除某条rcu发来的某个对象信息

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/rcu/obj

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| objsflag | 是 | String |  | rcu发来的信息唯一标志 |
| objId | 是 | String |  | 该消息中第几个对象 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | Varchar | 用于提示是否删除成功  1. success  2. fail |
| msg | 否 | Varchar | 1. 删除成功 2. 删除失败 |

**返回示例**

删除成功

{

"status": “success”,

“msg”: “删除成功”

}

删除失败

{

"status": “fail”,

“msg”：“删除失败”

}

### 根据rcuids和时间范围导出objs数据

**接口描述**

选择起始时间，终止时间，rcuIds然后导出rcuObj信息为xlsx

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/rcu/exportByRcuId

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| rcuIds | 是 | String |  | Rcuid列表，用逗号隔开的字符串 |
| startTime | 是 | string |  | 开始时间(时间戳) |
| endTime | 是 | string |  | 结束时间(时间戳) |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | success 成功  fail 失败 |
| filepath | 否 | string | 保存成功的下载路径 |

**返回示例**

保存成功

{

"status": "fail",

"filepath": "upload/xlsxs/001.xlsx"

}

保存失败

{

"status": "fail"

}

### 根据objsflags和时间范围导出objs数据

**接口描述**

选择起始时间，终止时间，objsflags然后导出rcuObj信息为xlsx

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/rcu/exportByobjsflag

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| objsflags | 是 | String |  | objsflag列表，用逗号分割的字符串 |
| startTime | 是 | string |  | 开始时间(时间戳) |
| endTime | 是 | string |  | 结束时间(时间戳) |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | success 成功  fail 失败 |
| filepath | 否 | string | 保存成功的下载路径 |

**返回示例**

保存成功

{

"status": "fail",

"filepath": "upload/xlsxs/001.xlsx"

}

保存失败

{

"status": "fail"

}

# 操作日志管理

## 4.1 页查询用户操作日志

**接口描述**

获取数据库日志，返回日志数据

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/log/list

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| pageNum | 否 | Int |  | 页码默认为1 |
| pageSize | 否 | Int |  | 每页大小默认为10 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| total | 是 | string | 查询到的总数量 |
| rows | 是 | list | 查询到的日志数据列表 |

**返回示例**

查询成功

{

"total": 15,

"rows": [

{

"oper\_time": "2025-04-11 10:58:05",

"error\_msg": "",

"method": "com.RoadCloudVisualizationSystem.controller.system.SystemUserController.getUserList()",

"oper\_param": "{\"pageSize\":[\"10\"],\"pageNum\":[\"1\"]}",

"oper\_name": "匿名用户",

"request\_method": "GET",

"title": "获取用户信息",

"oper\_ip": "192.168.120.64",

"json\_result": "{\"total\":21,\"rows\":[{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"2\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$BQUQc8h6imHaXG7y0.onYuG/i3KLyAusM1qwqg7o0wOB9W5huBeGW\",\"uid\":\"112323r24t54\_1744251978775\",\"username\":\"112323r24t54\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"1\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$GdyG44xDczgq9TFeRY8v0.YsEAM8E3/yUh859UGnxqSUPhHvVPMXm\",\"uid\":\"1231\_1744249943774\",\"username\":\"1231\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"2\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$E//F71Y.o4HxrulnJoMub.jLPySFHav0gtlJo./yD8/SEYUHZZ1qy\",\"uid\":\"abcd\_1744254569711\",\"username\":\"abcd\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"admin\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$iLDj7VVWV/nVdXlvlvT9XOBgNaKzI0XhUyRUtwrIfDkoQOBqT07ou\",\"uid\":\"admin\_1744335401465\",\"username\":\"admin\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"1\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$h6XxsMZuaZTG/eLDYIGGRuUDxFcTEnC21htynQR2jPOnQ8XktZ23a\",\"uid\":\"dsafds\_1744190387573\",\"username\":\"dsafds\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"2\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$U6cSfoNaAG.agIRbnO6mYuQPv3ZgQRfre1i3cGCgwATT5Kox1mHda\",\"uid\":\"dsfadsfa\_1744190394449\",\"username\":\"dsfadsfa\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"1\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$UZL7NsnvjS4nbLUG6frCkew3LAPBsN3ctyqEFshDOSVNt4qu4jbDG\",\"uid\":\"dsgffd\_1744194348936\",\"username\":\"dsgffd\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"1\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$P0Fn52YCat59JcRXd5vTf.dLX5BS0LwJdn0w0cEN.di7LO7NX8scO\",\"uid\":\"fgfghrgdhgfgsfd\_1744184758895\",\"username\":\"fgfghrgdhgfgsfd\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"1\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$YvnV3qj.2qr1DbyweFrzJu3m/tNh/HZ4yMjT2SFusPCRuGwYi9bL.\",\"uid\":\"ghtrhtue3rfl\_1744185345243\",\"username\":\"ghtrhtue3rfl\"},{\"accountNonExpired\":true,\"accountNonLocked\":true,\"authorities\":[],\"authority\":\"1\",\"credentialsNonExpired\":true,\"enabled\":true,\"password\":\"$2a$10$GPxhEscJwtbt0DiSATXMkOiN7yGuSSgop2qi9c14xe6VILmRbaSGC\",\"uid\":\"gsgfdg\_1744185278836\",\"username\":\"gsgfdg\"}],\"status\":\"success\"}",

"execute\_time": 100,

"business\_type": "OTHER",

"oper\_url": "/system/user/list",

"id": 67,

"status": 0

}

],

"status": "success"

}

查询失败

{

"status": “fail”,

}

## 删除数据库日志数据

**接口描述**

删除数据库的日志数据，通过其id(主键)

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/log/delete

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| id | 是 | int |  | 日志数据的主键 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

删除成功

{

"status": “success”,

}

删除失败

{

"status": “fail”,

}

## 导出数据库日志数据

**接口描述**

选择起始时间，结束时间，将对应日志数据导出

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/log/export

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| starttime | 否 | string |  | 起始时间(可不选，但不可不填) |
| endtime | 否 | string |  | 终止时间(可不选，但不可不填) |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

保存成功

{

"filepath": "/exports/Log\_匿名1744364939960.xlsx",

"status": "success"

}

保存失败

{

"status": "fail"

}

## 清空日志

**接口描述**

清空全部日志

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/log/clean

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

清空成功

{

"status": "success"

}

清空失败

{

"status": "fail"

}

# 系统参数管理

## 分页查询系统参数列表

**接口描述**

获取数据库中的系统参数列表

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/parameter/list

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| pageNum | 是 | Int |  | 页码默认为1 |
| pageSize | 是 | Int |  | 每页大小默认为10 |
| group | 否 | String |  | 组别 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| total | 是 | string | 查询到的总数量 |
| rows | 是 | list | 查询到的参数列表 |

**返回示例**

查询成功

{

"total": 15,

"rows": [

{

          id: 7,

          name: '系统名称',

          key: 'system.name',

          value: 'RoadCloud 交通系统',

          description: '系统显示名称',

          group: 'style'

        },

        {

          id: 9,

          name: '系统LOGO',

          key: 'system.logo.url',

          value: '@/assets/logo.png',

          description: '系统LOGO图片路径（使用@表示src目录）',

          group: 'style'

        }

      ]

"status": "success"

}

查询失败

{

"status": “fail”,

}

## 删除参数数据

**接口描述**

删除数据库的参数数据，通过其id(主键)

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/parameter

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| id | 是 | int |  | 日志数据的主键 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

删除成功

{

"status": “success”,

}

删除失败

{

"status": “fail”,

}

## 修改参数

**接口描述**

提供参数修改的功能，需要携带一个参数对象，其中必选包含id

**请求说明**

请求示例

HTTP 方法: PUT

请求URL: /system/parameter

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| parameter | 是 | Parameter |  | 参数对象，其中必须包含id（id不可修改） |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示更新成功或者失败  1. success  2. fail |

**返回示例**

保存成功

{

"status": "success"

}

保存失败

{

"status": "fail"

}

## 新增参数

**接口描述**

提供新增的功能，需要携带一个参数对象，不需要包含id

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /system/parameter

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| parameter | 是 | Parameter |  | 参数对象 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示插入成功或者失败  1. success  2. fail |

**返回示例**

保存成功

{

"status": "success"

}

保存失败

{

"status": "fail"

}

# 首页界面

## 交通设备数量总览-查询

**接口描述**

本接口实现了首页界面的实时设备信息总览,包括rcu,摄像头，雷达，激光雷达。其中rcu是代表所有设备的总数。要求前端定时发起请求以实时更新。后端需要根据数据库中rcu表中deviceType字段写一个枚举类，以便直接返回前端设备名称（中文？）。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/device/numList

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示查询成功或者失败  1. success  2. fail |
| deviceNumList | 是 | List<Map<String,Integer>> | 查询到的日志数据列表 |

**返回示例**

查询成功

{

"status": “success”,

"deviceNumList": [

{“rcu”:1},

{“摄像头”:1},

{“融合设备”:1},

{“lidar”:1},

]

}

查询失败

{

"status": “fail”,

}

## 设备健康状态分布-查询

**接口描述**

本接口实现了根据设备类型查询异常设备，返回各种状态的设备数量。异常设备是指某设备在上次发送消息之后的60s内没有再次发送消息，离线设备是指某设备在上次发送消息之后的3600s内没有再次发送消息（实际状态信息参考路云标准9.4表25-31）。

（真实测试时可能没有异常设备，需要后端手动插入两个不会更新的设备）

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/device/status

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| deviceStatus | 否 | Map<String,List<Integer>> | 以设备类型进行分类，每个设备类型中的List[0]代表正常数量，List[1]代表告警数量，List[2]代表离线数量 |

**返回示例**

查询成功

{

"status": “success”,

“deviceStatus”: {

“摄像头”: [7, 2, 1],

“融合设备”:[3, 5, 7]

}

}

查询失败

{

"status": “fail”

}

## 查询异常设备（告警、离线）

**接口描述**

本接口实现了查询异常设备，返回异常设备ID列表。告警设备是指某设备在上次发送消息之后的60s内没有再次发送消息（实际状态信息参考路云标准9.4表25-31）。离线设备是指某设备在上次发送消息之后的3600s内没有再次发送消息

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/device/abnormal

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| total | 否 | Integer | 异常设备总数量 |
| abnormalDevices | 否 | List<Map<String,String>> | 离线设备类型和ID列表 |

**返回示例**

查询成功

{

"status": “success”,

“total”: 2，

“abnormalDevices”:[

{

“latestTime”: 159848153134, (设备最后响应时间)

“rcuId”: “q-sadsv”,

“deviceType”: “摄像头”,

“abnormalType”: “告警”

},

{

“latestTime”: 159848153134, (设备最后响应时间)

“rcuId”: “q-sadsv”,

“deviceType”: “摄像头”,

“abnormalType”: “离线”

}

]

}

查询失败

{

"status": “fail”

}

## 根据RcuId查询设备信息

**接口描述**

本接口实现了根据RcuId查询设备信息和状态的功能

应用场景：通过搜索或者点击触发rcuId查询设备信息。



**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/device

请求参数: 无

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| rcuId | 是 | String | 设备id |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| msg | 否 | String | 1. 设备不存在 2. 服务器内部错误 |
| device | 否 | Rcu | 设备信息 |
| deviceStatus | 否 | String | 设备状态 |

**返回示例**

查询成功

{

    "device": {

        "rcuId": "U-WZ0012",

        "deviceType": "摄像头",

        "receiveTime": "1744438020507",

“longitude”: “106.5312”,

“latitude”: “29.4884”

    },

    "status": "success",

    "deviceStatus": "离线"

}

查询失败

{

"status": “fail”,

“msg”: “设备不存在”

}

## 根据车辆id查询车辆信息

**接口描述**

本接口实现了根据车辆id查询车辆，返回指定的车辆信息列表。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle

请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| vehicleId | 是 | String |  | 车辆ID |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| vehicle | 否 | Vehicle | 车辆当前状态信息 |
| trackPoints | 否 | List<Map> | 历史轨迹点信息 |

**返回示例**

查询成功

{

"status": “success”,

“vehicle”:{

" vehicleId ": " QD1E000J",

“longitude”:180.0000000,（最新经度）

“latitude”:29.00000000,（最新纬度）

“count”:100,(总的轨迹点数量)

"latestTime": "1723676700778"（最新时间）

}

"trackPoints": [

        {

            "latitude": "29.6299708",

            "timestampGNSS": "1723498400532",

            "longitude": "106.3281523"

        },

        {

            "latitude": "29.6299708",

            "timestampGNSS": "1723498400732",

            "longitude": "106.3281523"

        }

}

查询失败

{

"status": “fail”

}

## 拓扑图填充

**接口描述**

本接口实现了对于拓扑图的填充，返回重庆市中心经纬度，以及其分支下各个区县经纬度，然后路口经纬度，路口经纬度对于设备经纬度

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/top/list

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| provincename | 是 | string |  | 省份拼音，重庆(chongqing)、广东两个省份 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示查询成功或者失败  1. success  2. fail |
| name | 否 | string | 省份名称 |
| data | 否 | string | 数据详细 |

**返回示例**

查询成功

{

"provicename": "重庆市",

"data": {

"provicename": "重庆市",

"lng": 106.551557,

"lat": 29.563009,

"children": [

{

"city": "渝中区",

"lng": 106.571746,

"lat": 29.557404,

"children": [

{

"landmark": "解放碑",

"lng": 106.573347,

"lat": 29.558964,

"children": [

{

"rcuId": "U-WZ000R",

"lng": 106.5736,

"lat": 29.5592

},

{

"rcuId": "U-WZ000S",

"lng": 106.5737,

"lat": 29.5593

}

]

},

{

"landmark": "临江",

"lng": 106.578185,

"lat": 29.562458,

"children": [

{

"rcuId": "U-WZ0012",

"lng": 106.5784,

"lat": 29.5627

},

{

"rcuId": "U-WZ0013",

"lng": 106.5786,

"lat": 29.5629

},

{

"rcuId": "U-WZ0015",

"lng": 106.5783,

"lat": 29.5625

}

]

}

]

},

{

"city": "沙坪坝区",

"lng": 106.463656,

"lat": 29.555125,

"children": [

{

"landmark": "三峡广场",

"lng": 106.463021,

"lat": 29.555384,

"children": [

{

"rcuId": "U-WZ0019",

"lng": 106.4633,

"lat": 29.55555

},

{

"rcuId": "U-WZ001B",

"lng": 106.4634,

"lat": 29.5556

}

]

},

{

"landmark": "谢家湾",

"lng": 106.46244,

"lat": 29.552685,

"children": [

{

"rcuId": "U-WZ004K",

"lng": 106.4627,

"lat": 29.55295

}

]

}

]

},

{

"city": "九龙坡区",

"lng": 106.530658,

"lat": 29.487746,

"children": [

{

"landmark": "石坪桥",

"lng": 106.53084,

"lat": 29.48806,

"children": [

{

"rcuId": "U-ZS000U",

"lng": 106.5311,

"lat": 29.4883

},

{

"rcuId": "U-ZS0013",

"lng": 106.5312,

"lat": 29.4884

}

]

}

]

}

]

},

"status": "success"

}

查询失败

{

"status": “fail”,

}

## 首页流量分析曲线图

**接口描述**

通过时间范围来统计所有路口的各个时间段的车流量

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/stream/all

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| starttime | 是 | string |  | 开始时间 |
| enditime | 是 | string |  | 结束时间 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| data | 否 | json | 有100个时间段的车流量 |

**返回示例**

查询成功

{

    "data": [

        {

            "start\_time": "2024-08-19T02:30:55.275+00:00",

            "seg\_start": 1.724034655275E12,

            "count": 2,

            "end\_time": "2024-08-19T02:30:56.274+00:00",

            "seg\_end": 1.724034656274E12

        },

]

}

查询失败

{

"status": “fail”,

}

## 首页根据时间范围统计车辆类型分布-饼图

**接口描述**

通过时间范围来统计所有路口的各个种类车辆的数量

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/kindnum/all

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| startTime | 是 | string |  | 开始时间戳 |
| endTime | 是 | string |  | 结束时间戳 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| data | 否 | List<Map> | 对应时间的车流量 |

**返回示例**

查询成功

{

"rcuId": "12345",

"data": [

{

“type”:”摩托车”,

"count": 350,

"percentage": 0.58

},

{

“type”: “自行车”,

"count": 120,

"percentage": 0.20

}

]

}

查询失败

{

"status": “fail”

}

## 获取交通事件

**接口描述**

本接口实现了查询交通事件的功能，每次请求后端会自动更新需要展示的交通事件。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/device/event

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| total | 否 | Integer | 交通事件总数 |
| events | 否 | List<Event> | 离线设备类型和ID列表 |

**返回示例**

查询成功

{

    "total": 3,

    "events": [

        {

            "eventId": "E001",

            "eventType": 1,

            "rcuId": "U-ZS000U",

            "longitude": "106.5312",

            "latitude": "29.4884",

            "timestamp": "1744558689190",

            "targetIdsLen": 0,

            "targetIds": "",

            "intersectionName": "石坪桥路口2",

            "isView": 0,

            "description": "慢行"

        },

        {

            "eventId": "E005",

            "eventType": 2,

            "rcuId": "U-WZ0019",

            "longitude": "106.4634",

            "latitude": "29.5556",

            "timestamp": "1744558689190",

            "targetIdsLen": 0,

            "targetIds": null,

            "intersectionName": "三峡广场路口1",

            "isView": 0,

            "description": "快行"

        },

        {

            "eventId": "E008",

            "eventType": 3,

            "rcuId": "U-WZ0012",

            "longitude": "106.5784",

            "latitude": "29.5627",

            "timestamp": "1744558689190",

            "targetIdsLen": 0,

            "targetIds": null,

            "intersectionName": "临江路口1",

            "isView": 0,

            "description": "紧急制动"

        }

    ],

    "status": "success"

}

查询失败

{

"status": “fail”

}

## 统计人、车数量和差值

**接口描述**

本接口实现了根据时间戳统计人、车总量以及差值的功能，需要前端传入一个当前时刻的时间戳，后台会统计

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/count

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| timestamp | 是 | string |  | 当前时刻时间戳 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| vehicleCount | 否 | Integer | 车辆总数 |
| vehicleDiff | 否 | Integer | 与前一时刻差值 |
| pedestrianCount | 否 | Integer | 行人总数 |
| pedestrianDiff | 否 | Integer | 与前一时刻差值 |

**返回示例**

查询成功

{

"status": “success”,

    "vehicleDiff": 41,

    "pedestrianDiff": 0,

    "vehicleCount": 41,

    "pedestrianCount": 0,

    "status": "success"

}

查询失败

{

"status": “fail”

}

## 地标路段实时流量排名

**接口描述**

本接口实现了对各个路口的流量进行流量统计和实时排名，需要提供当前时间的时间戳，返回一个排好序的列表，里面有路口名称和流量大小。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/stream/rank

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| timestamp | 是 | string |  | 时间戳 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| streams | 否 | List<Map> | 排好序的流量列表，里面有区域名称、流量大小 |

**返回示例**

查询成功

{

"status": “success”

   “streams”: [

{

“landmark”: “解放碑”,

“stream”: 175

},

{

“landmark”: “临江”,

“stream”: 185

}

]

}

查询失败

{

"status": “fail”

}

## 分段统计车辆类型以及数量列表

**接口描述**

本接口实现了对所有路口的不同类型的车辆类型分段统计数量。

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/carType/count

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| startTime | 是 | string |  | 时间戳 |
| endTime | 是 | string |  | 时间戳 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| carTypeAndCount | 否 | Map <Map> | 各类型的数据列表 |

**返回示例**

查询成功

{

    "carTypeAndCount": {

        "乘用车": [16,10,2,27,34,25,17,0],

        "自行车": [0,0,0,0,2,7,0,0,7,…],

        "卡车": [3,2,0,2,0,0,0,0,4,… ]

    },

    "status": "success"

}

查询失败

{

"status": “fail”

}

## 统计车辆速度分布区间和数量

**接口描述**

本接口实现了对车辆速度区间分布的统计

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/speed

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| vehicleSpeedRange | 否 | List<Map<String,Object>> | 分布区间速度、数量列表 |

**返回示例**

查询成功

{

    " vehicleSpeedRange ": [

        {

            "speed\_range": "2-4 m/s",

            "vehicle\_count": 1

        },

        {

            "speed\_range": "8+ m/s",

            "vehicle\_count": 2

        }

    ],

    "status": "success"

}

查询失败

{

"status": “fail”

}

## 分页查询设备列表

**接口描述**

本接口实现了分页查询设备列表的功能

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/device/list

请求参数: 无

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| pageNum | 是 | Int | 页码 |
| pageSize | 是 | Int | 每页大小 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| total | 否 | Int | 设备总数量 |
| rows | 否 | List<Rcu> | 设备列表 |

**返回示例**

查询成功

{

    "total": 10,

    "rows": [

        {

            "rcuId": "U-WZ000R",

            "deviceType": "2",

            "receiveTime": "1744700893878",

            "longitude": "106.5736",

            "latitude": "29.5592"

        },

        {

            "rcuId": "U-WZ000S",

            "deviceType": "2",

            "receiveTime": "1744700892799",

            "longitude": "106.5737",

            "latitude": "29.5593"

        },

        {

            "rcuId": "U-WZ0012",

            "deviceType": "2",

            "receiveTime": "1744700894211",

            "longitude": "106.5784",

            "latitude": "29.5627"

        }

    ],

    "status": "success"

}

查询失败

{

"status": “fail”

}

## 分页查询车辆列表

**接口描述**

将车辆根据vehicleId进行分类，分别包含其对应的位置点数量(position)，以及最后更新时间

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/list

Parma请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| pagaNum | 否 | String | 从1开始 | 为空默认为1 |
| pageSize | 否 | String |  | 为空默认为10 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | success 成功  fail 失败 |
| total | 是 | Int | 数据总条数 |
| rows | 是 | List | 所有车的列表信息 |

**返回示例**

查询成功

{

    "total": 12,

    "rows": [

        {

            "formatted\_time": "2025-04-16T05:32:31.960+00:00",

            "timestampGNSS": "1744781551960",

            "velocityMS": "3.74",

            "vehicleId": "QD1E002C",

            "velocityKMH": 13.46

        },

        {

            "formatted\_time": "2025-04-16T05:32:31.848+00:00",

            "timestampGNSS": "1744781551848",

            "velocityMS": "0.0",

            "vehicleId": "QD1E000R",

            "velocityKMH": 0.00

        },

        {

            "formatted\_time": "2025-04-16T05:32:31.557+00:00",

            "timestampGNSS": "1744781551557",

            "velocityMS": "0.0",

            "vehicleId": "QD1E0026",

            "velocityKMH": 0.00

        }

    ],

    "status": "success"

}

查询失败

{

"status": "fail"

}

# 详细页界面

## 详细页流量分析曲线图

**接口描述**

一个rcuId即对应一个路口，通过rcuId、和日期来统计该路口的各个时间段的车流量

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/stream

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| rcuId | 是 | string |  | rcu设备Id |
| startTime | 是 | string |  | 开始时间 |
| endTime | 是 | string |  | 结束时间 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| data | 否 | json | 有100个时间段的车流量 |

**返回示例**

查询成功

{

    "data": [

        {

            "start\_time": "2024-08-19T02:30:55.275+00:00",

            "seg\_start": 1.724034655275E12,

            "count": 2,

            "end\_time": "2024-08-19T02:30:56.274+00:00",

            "seg\_end": 1.724034656274E12

        },

]

}

查询失败

{

"status": “fail”,

}

## 详细页流量分析条形图

**接口描述**

一个rcuId即对应一个路口，通过rcuId、和日期来统计该路口的各个种类车辆的数量

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/vehicle/kindnum

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| rcuId | 是 | string |  | rcu设备Id |
| starttime | 是 | string |  | 时间戳 |
| endtime | 是 | string |  | 时间戳 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| data | 否 | List<Map> | 对应时间的车流量 |

**返回示例**

查询成功

{

"rcuId": "12345",

"data": [

{

“type”:”摩托车”,

"count": 350,

"percentage": 0.58

},

{

“type”: “自行车”,

"count": 120,

"percentage": 0.20

}

]

}

查询失败

{

"status": “fail”

}

## 查询所有可建模数据--Blender相关

**接口描述**

根据页数和每页页数大小，访问数据库，并返回相应的可用来建模的数据

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/creat/list

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| pageNum | 是 | string |  | 页数 |
| pageSize | 是 | string |  | 每页大小 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| data | 否 | List<Map> | 对应的数据 |

**返回示例**

查询成功

{

"data": [

{

"elevation": "0.0",

"times": "9",

"obj\_id": "car1",

"color": "pink",

"heading": "180.0",

"series": "0",

"latitude": "29.5106444",

"type": "vehicle",

"longitude": "106.4686432"

},

{

"elevation": "0.0",

"times": "9",

"obj\_id": "car13",

"color": "blue",

"heading": "-90.0",

"series": "0",

"latitude": "29.5101127",

"type": "vehicle",

"longitude": "106.4693553"

},

{

"elevation": "0.0",

"times": "9",

"obj\_id": "car4",

"color": "black",

"heading": "136.0",

"series": "0",

"latitude": "29.5100410",

"type": "vehicle",

"longitude": "106.4683789"

},

{

"elevation": "0.0",

"times": "9",

"obj\_id": "car2",

"color": "blue",

"heading": "90.0",

"series": "0",

"latitude": "29.5098254",

"type": "vehicle",

"longitude": "106.4695658"

}

]

statu: "success"

}

查询失败

{

"status": “fail”

}

## 导出建模数据

**接口描述**

根据起始时间，终止时间，输出为相应的csv文件格式方便后续用来建模

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/creat/export

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| starttime | 是 | string |  | 起始时间 |
| endtime | 是 | string |  | 终止时间 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| filepath | 否 | List<Map> | 导出路径 |

**返回示例**

导出成功

{

"filepath": "/exports/1744608747530.csv",

"status": "success"

}

导出失败

{

"status": “fail”

}

## 建模数据转模型文件

**接口描述**

上传csv格式文件数据，后台进行转换并返回下载连接

**请求说明**

请求示例

HTTP 方法: Post

请求URL: 106.12.131.107:5147

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| file | 是 | file |  | csv格式文件 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| filepath | 否 | List<Map> | 下载连接 |

**返回示例**

转换成功

{

"filepath": "/output/1744608747530.glb",

"status": "success"

}

转换失败

{

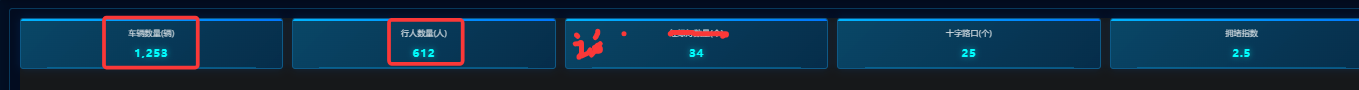
"status": “fail”

}

## 根据RcuId路口信息

**接口描述**

本接口实现了根据RcuId路口信息，具体包括：车流量、人流量、设备类型、设备状态、设备最久更新时间。



**请求说明**

请求示例

HTTP 方法: GET

请求URL: /menu/device/rcuInfo

请求参数: 无

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| rcuId | 是 | String | 设备id |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | String | 用于提示是否查询成功  1. success  2. fail |
| rcuInfo | 否 | Map | 设备相关信息 |

**返回示例**

查询成功

{

    "rcuInfo": {

        "deviceType": "摄像头",

        "last\_update\_time": "1744709437664",

        "rcuId": "U-WZ000R",

        "last\_update\_time\_formatted": "2025-04-15 17:30:37.664000",

        "person\_count": 255,

        "vehicle\_count": 1865,

        "status": "离线"

    },

    "status": "success"

}

查询失败

{

"status": “fail”,

}

# RBAC权限管理

## 查询所有角色

**接口描述**

查询该系统有哪些角色，查看每个角色基本信息

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/role

Parma请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| pageNum | 是 | string |  | 页数 |
| pageSize | 是 | string |  | 每页大小 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |
| total | 否 |  | 总数据条数 |
| rows | 否 |  | 数据 |

**返回示例**

查询成功

{

" status":" success" ,

    " total ": 3,

    " rows ": [

        {

            "rolecode": "ROLE\_USER",

            "roleid": "1",

            "rolename": "普通用户"

        },

        {

            "rolecode": "ROLE\_ADMIN",

            "roleid": "2",

            "rolename": "管理员"

        },

        {

            "rolecode": "ROLE\_SUPERADMIN",

            "roleid": "3",

            "rolename": "超级管理员"

        }

    ]

}

查询失败

{

"status": "fail"

}

## 创建角色

**接口描述**

为系统创建一个角色，其权限为空，没有任何

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /system/role

Body请求参数(json):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| rolecode | 是 | string |  | 角色的英文名必须全大写且ROLE\_XXX 格式 |
| rolename | 是 | string |  | 角色的中文名，管理员可以看懂的 |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

添加成功

{

"status": "success"

}

添加失败

{

"status": "fail"

}

## 删除角色

**接口描述**

输入roleid ，删除角色

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/role

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| roleId | 是 | string |  | 角色的Id |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

删除成功

{

"status": "success"

}

删除失败

{

"status": "fail"

}

## 查询所有权限

**接口描述**

获取所有权限的描述以及对应的id

**请求说明**

请求示例

HTTP 方法: GET

请求URL: /system/role/permission

请求参数: 无

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

添加成功

{

"status": "success"

}

添加失败

{

"status": "fail"

}

## 增加角色权限

**接口描述**

输入roleid,permissiondata(多个permissionid用 , 连接)增加角色权限

**请求说明**

请求示例

HTTP 方法: POST

请求URL: /system/role/permission

Body请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| roleId | 是 | string |  | 角色的Id |
| permissiondata | 是 | string |  | 用 , 拼接多个权限的id |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

添加成功

{

"status": "success"

}

添加失败

{

"status": "fail"

}

## 删除角色权限

**接口描述**

输入roleid,permissiondata(多个permissionid用 , 连接)删除角色权限

**请求说明**

请求示例

HTTP 方法: DELETE

请求URL: /system/role/permission

Param请求参数:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| roleId | 是 | string |  | 角色的Id |
| permissiondata | 是 | string |  | 用 , 拼接多个权限的id |

**返回说明**

返回参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 是否必选 | 类型 | 说明 |
| status | 是 | string | 用于提示查询成功或者失败  1. success  2. fail |

**返回示例**

添加成功

{

"status": "success"

}

添加失败

{

"status": "fail"

}